

БЕСШОВНЫЙ СИНТЕЗ СВЕРХБОЛЬШИХ ТЕКСТУР ПЛАНЕТ

А.В.МАЛЬЦЕВ, П.Ю.ТИМОХИН

Научно-исследовательский институт системных исследований РАН
Москва, Россия
e-mail: avmaltcev@mail.ru, webpismo@yahoo.de

Ключевые слова: Сверхбольшая текстура, визуализация, планета

Аннотация. В статье предложена технология подготовки сверхбольшой текстуры поверхности планеты. Технология позволяет синтезировать такую текстуру по частям, обеспечивая при этом бесшовную стыковку ее результирующих фрагментов. Для повышения общей скорости синтеза сверхбольшой текстуры в работе предлагается система параллельного вычисления частей текстуры на ряде компьютеров, объединенных в быструю сеть.

SEAMLESS SYNTHESIS OF EXTRA LARGE PLANET TEXTURES

A.V. MALTSEV, P.YU. TIMOKHIN

Scientific Research Institute for System Analysis of RAS
Moscow, Russia
e-mail: avmaltcev@mail.ru, webpismo@yahoo.de

Summary. The paper proposes a technology for preparing of extra-large texture of planet surface. The technology allows synthesizing such texture by parts, while providing a seamless splicing of its final fragments. To improve the overall rate of extra-large texture synthesis we propose a system of parallel calculating of such texture parts on a number of computers connected to fast network.

2010 Mathematics Subject Classification: 94A08.

Key words and Phrases: Extra large Texture, Visualization, Planet.

1 ВВЕДЕНИЕ

Одной из актуальных задач современных космических видео тренажерных комплексов является визуализация детализированных трехмерных виртуальных поверхностей планет в режиме реального времени. С точки зрения соотношения скорости и качества визуализации эффективным способом повышения детализации виртуального ландшафта планеты является применение высоко детализированных текстур, созданных на основе массивов спутниковых снимков высокого разрешения¹.

В силу ряда факторов один такой массив может содержать изображение, покрывающее лишь часть поверхности планеты, поэтому для синтеза полной текстуры приходится привлекать массивы спутниковых изображений, полученные из нескольких источников. Как правило, форматы представления таких данных (разрешение, картографическая проекция и др.) отличаются от формата основного массива спутниковых изображений, вследствие чего возникает задача интеграции таких данных в текстуру планеты. Для достижения качественного визуального результата при такой интеграции необходимо обеспечивать отсутствие видимых швов в местах стыка основного и интегрируемого изображения. В полной мере бесшовная стыковка включает в себя стыковку изображений по цвету (отенок насыщенность, яркость) и по геометрическим характеристикам (углы, пропорции, масштаб и др.). Такая стыковка возможна с использованием информации из смежных областей текстуры.

В случае подготовки детализированной текстуры небольшого размера, такого рода интеграция данных может быть выполнена в общей памяти рабочей станции, где обеспечивается прямой доступ ко всем областям текстуры². Однако данный подход становится неприменимым при необходимости подготовки текстуры, размер и объем которой превышают аппаратные возможности отдельной рабочей станции (далее будем называть такие текстуры *сверхбольшими*). Логичным путем обхода этого ограничения является подготовка сверхбольшой текстуры по частям с последующим объединением готовых фрагментов в единое изображение, размещаемое в хранилище данных. Эффективность синтеза сверхбольшой текстуры может быть существенно повышена путем распределения подготовки фрагментов текстуры между несколькими вычислительными узлами, ввиду чего возникает задача разработки такой технологии, которая бы позволяла подготавливать сверхбольшую текстуру по частям, независимым друг от друга, и обеспечивала бесшовную стыковку готовых фрагментов в результирующей текстуре.

В данной работе предлагается технология такой подготовки, позволяющая интегрировать изображения, заданные в распространенной проекции Меркатора, в единую текстуру, заданную в эквидистантной цилиндрической проекции.

2 ПОСТАНОВКА ЗАДАЧИ

Исходными данными в рассматриваемой задаче являются: текстура M поверхности планеты, заданная в проекции Меркатора, размера $W_M \times H_M$ пикселей ($W_M = H_M = 2^m$, $m > h$, где 2^h - максимальный размер текстуры, обрабатываемый отдельной рабочей станцией), текстура D поверхности планеты размера $W_D \times H_D$ пикселей ($W_D = 2^{m-k}$, $H_D = 2^{m-k-1}$, $m - k \geq h$) и маска G поверхности планеты размера $W_G \times H_G$

пикселей ($W_G = 2^{m-g}$, $H_G = 2^{m-g-1}$, $m-g \geq h$), заданные в эквидистантной цилиндрической проекции (в такой проекции высота изображения в 2 раза меньше ширины³).

Текстура M содержит высоко детализированное изображение большей части поверхности планеты, созданное на основе спутниковых снимков. В силу математической особенности проекции Меркатора, данная текстура не охватывает приполярные области планеты. Кроме того, текстура M содержит пустые регионы, где спутниковая съемка не проводилась или была прервана по тем или иным причинам. Текстуры с такой высокой детализацией появляются в результате работы многолетних космических программ по дистанционному исследованию планет и формируются и пополняются в течение достаточно продолжительного периода времени.

Текстура D содержит изображение поверхности планеты, отснятое полностью, но с меньшей детализацией, чем у текстуры M . Примером такой текстуры может служить текстура земной поверхности из серии Blue Marble (NASA).

Маска G охватывает всю поверхность планеты и задана в градациях серого цвета, где черным и белом цветом (0.0 и 1.0) соответственно отмечены пустые и отснятые регионы в текстуре M , а серым цветом, плавно меняющимся от 0.0 до 1.0, – области перехода между этими регионами, отступающие от границы перехода вглубь отснятых регионов. Необходимая для создания маски G битовая информация о пустых и отснятых регионах планеты извлекается непосредственно из текстуры M , а также из доступных масок, получаемых космическими агентствами в качестве побочного продукта предварительной обработки спутниковых изображений. Примером такой вспомогательной маски является битовая маска водных поверхностей, поставляемая агентством NASA вместе с текстурами земной поверхности серии Blue Marble. После внесения в маску G битовых данных выполняется добавление описанных выше областей плавного перехода с помощью разработанной ранее технологии модификации битовой маски². В общем случае детализация маски G может быть меньше, чем у текстуры M , и отличается от детализации текстуры D .

Наша задача состоит в вычислении текстуры E размера $W_E \times H_E$ ($W_E = 2^m$, $H_E = 2^{m-1}$), соответствующей текстуре M , но заданной в эквидистантной цилиндрической проекции. Цвет каждого пикселя текстуры E будет вычисляться: а) из текстуры M в тех участках, где маска G заполнена белым цветом, б) из текстуры D , где маска G заполнена черным цветом и в) из текстур M и D , в тех местах, где маска G заполнена серым цветом. В работе² описан подобный метод расчета цвета каждого пикселя текстуры E в задаче синтеза детальной текстуры земной поверхности. Ввиду того, что текстуры E , M , D и G в данной задаче являются сверхбольшими, вычисление текстуры E должно выполняться по группам пикселей (тайлам), которые могут эффективно, независимо друг от друга и параллельно обрабатываться на отдельных вычислительных узлах (компьютерах). Далее рассмотрим алгоритм расчета цветов для тайла текстуры E , а также подход к организации вычисления всех таких тайлов на ряде вычислительных узлов, объединенных в быструю сеть.

3 ВЫЧИСЛЕНИЕ ТАЙЛА РЕЗУЛЬТИРУЮЩЕЙ ТЕКСТУРЫ E

Пусть исходные текстуры M , D и G разбиты на тайлы с размерами $W_{T,M} \times H_{T,M}$, $W_{T,D} \times H_{T,D}$ и $W_{T,G} \times H_{T,G}$ пикселей соответственно. Синтезируемая текстура E также будет состоять из тайлов, имеющих размер $W_{T,E} \times H_{T,E}$ пикселей. Далее будем считать, что индексация тайлов во всех текстурах проводится от левого нижнего угла.

Рассмотрим процесс подготовки тайла T_E текстуры E с координатами $(i_{T,E}, j_{T,E})$. Вначале рассчитаем координаты $(i_{E,min}, j_{E,min})$ и $(i_{E,max}, j_{E,max})$ соответственно левого нижнего и правого верхнего пикселей тайла T_E в системе координат $I_E J_E$ пикселей с началом в центре текстуры E (см. ²) по формулам:

$$\begin{aligned} i_{E,min} &= i_{T,E}W_{T,E} - W_E / 2, & j_{E,min} &= j_{T,E}H_{T,E} - H_E / 2; \\ i_{E,max} &= (i_{T,E} + 1)W_{T,E} - W_E / 2, & j_{E,max} &= (j_{T,E} + 1)H_{T,E} - H_E / 2. \end{aligned} \quad (1)$$

Далее сформируем из некоторого подмножества исходных тайлов текстуры G тайл маски T_G с размером $W_{T,E} \times H_{T,E}$ пикселей, имеющий то же пространственное разрешение (в метрах на пиксел) и охватывающий тот же регион планеты, что и T_E . Для этого вычислим коэффициент k_G отношения разрешений текстур G и E как W_G / W_E . Тогда граничным пикселям $(i_{E,min}, j_{E,min})$ и $(i_{E,max}, j_{E,max})$ тайла T_E будут соответствовать пиксели с координатами

$$\begin{aligned} (i_{G,min}, j_{G,min}) &= (\lfloor k_G i_{E,min} \rfloor, \lfloor k_G j_{E,min} \rfloor), \\ (i_{G,max}, j_{G,max}) &= (\lfloor k_G i_{E,max} \rfloor, \lfloor k_G j_{E,max} \rfloor) \end{aligned}$$

в текстуре G (система координат $I_G J_G$ аналогична $I_E J_E$). Индексы исходных тайлов текстуры G , входящих в искомое подмножество, будут лежать между крайними левым нижним и правым верхним тайлами с координатами

$$\begin{aligned} &(\lfloor (i_{G,min} + W_G / 2) / W_{T,G} \rfloor, \lfloor (j_{G,min} + H_G / 2) / H_{T,G} \rfloor), \\ &(\lfloor (i_{G,max} + W_G / 2) / W_{T,G} \rfloor, \lfloor (j_{G,max} + H_G / 2) / H_{T,G} \rfloor) \end{aligned} \quad (2)$$

соответственно.

Полученную в (2) область из тайлов необходимо привести к такому же пространственному разрешению, как и у текстуры E . Это может быть качественно сделано с помощью какой-либо функции f масштабирования с учетом окрестностей пикселей, например, основанной на методе бикубической фильтрации. Заметим, что для каждого из тайлов T_G , формируемых в процессе подготовки различных тайлов текстуры E , такое преобразование будет выполняться независимо. Поэтому для избегания последующей нестыковки по краям отмасштабированных тайлов T_G необходимо, чтобы обрабатываемые независимо друг от друга соседние области текстуры G перекрывались на границах. Для этого в дополнение к определенным выше тайлам используем также «обрамление» толщиной в 1 тайл. Найденное с помощью формулы (2) подмножество тайлов текстуры G , включая «обрамление», образует текстуру G' .

Применим к G' функцию f , задав коэффициент масштабирования равным $1 / k_G$. Чтобы сформировать тайл T_G необходимо обрезать полученное после масштабирования

изображение G'' . Количество пикселей, удаляемых с каждой из четырех сторон G'' , определяется с учетом добавленного ранее «обрамления» из тайлов и, например, для нижней границы равно

$$\frac{1}{k_G} \left(H_{T,G} + (j_{G,\min} + H_G / 2) \% H_{T,G} \right),$$

где операция $\%$ означает «остаток от деления». Число обрезаемых с других сторон пикселей рассчитывается аналогично.

В зависимости от цветов пикселей сформированного тайла маски T_G возможны три различных варианта синтеза тайла T_E .

I. Если T_G содержит только черные пиксели (числовой эквивалент цвета каждого пиксела равен 0), это означает, что в исходной текстуре M нет данных для формирования T_E . Тогда необходимо из некоторого подмножества исходных тайлов дополнительной текстуры D синтезировать тайл T_D с размером $W_{T,E} \times H_{T,E}$ пикселей, имеющий то же пространственное разрешение и охватывающий тот же регион планеты, что и T_E . Данная процедура производится аналогично описанному выше формированию тайла T_G . При этом тайл T_E получим копированием данных из T_D .

II. Если T_G включает только белые пиксели (числовой эквивалент цвета каждого пиксела равен 1), то текстура M содержит всю необходимую информацию для синтеза тайла T_E . Чтобы сформировать T_E используем преобразование проекции Меркатора в эквидистантную проекцию. Так как M и E охватывают долготный диапазон $[-\pi, \pi]$, их ширина совпадает ($W_E = W_M = 2^m$), а масштаб карты M по оси ординат растет от экватора к полюсам³, то любому пикселу $P \in T_E$ с координатами (i_E, j_E) соответствует вертикальный отрезок R из некоторого числа r пикселей $(i_M, j_{M,k}) \in M$ (в системе координат $I_M J_M$ для индексирования пикселей, аналогичной $I_E J_E$), где $i_M = i_E$ и $k \in [0, r-1]$. Нахождение границ $j_{M,0}$ и $j_{M,r-1}$ для отрезка R подробно описано в работе² (см. формулу (4)).

Определим подмножество тайлов из карты M , которые содержат информацию для синтеза T_E . Для этого найдем отображение граничных пикселей $(i_{E,\min}, j_{E,\min})$ и $(i_{E,\max}, j_{E,\max})$ тайла T_E в текстуре M (координаты этих пикселей вычислены в формуле (1)). Используя описанное выше соответствие проекций, получим для $(i_{E,\min}, j_{E,\min})$ отрезок R_0 из r_0 пикселей $(i_{E,\min}, j_{M,u}^0) \in M$, $u \in [0, r_0-1]$, а для $(i_{E,\max}, j_{E,\max})$ - отрезок R_1 из r_1 пикселей $(i_{E,\max}, j_{M,v}^1) \in M$, $v \in [0, r_1-1]$. Тогда индексы тайлов текстуры M , входящих в искомое подмножество, будут лежать между крайними левым нижним и правым верхним тайлами с координатами

$$\begin{aligned} (i_{T,M,\min}, j_{T,M,\min}) &= \left(\left\lfloor (i_{E,\min} + W_M / 2) / W_{T,M} \right\rfloor, \left\lfloor (j_{M,0}^0 + H_M / 2) / H_{T,M} \right\rfloor \right), \\ (i_{T,M,\max}, j_{T,M,\max}) &= \left(\left\lfloor (i_{E,\max} + W_M / 2) / W_{T,M} \right\rfloor, \left\lfloor (j_{M,r_1-1}^1 + H_M / 2) / H_{T,M} \right\rfloor \right) \end{aligned} \quad (3)$$

соответственно. Найденное подмножество тайлов образует текстуру M' .

Цвет C_P пиксела $P \in T_E$ рассчитывается как среднее арифметическое цветов всех r пикселей $((i_M, j_{M,k}), k \in [0, r-1])$ из отрезка R , который является отображением P в карте

M . Чтобы отсчитывать пиксели от начала системы координат текстуры M' , выполним для R сдвиг начала системы координат $I_M J_M$ из центра текстуры M в левый нижний угол ее области M' . Тогда:

$$C_P = \frac{1}{r} \sum_{k=0}^{r-1} C_{M'}(i_M + 0.5W_M - i_{T,M,\min} W_{T,M}, j_{M,k} + 0.5H_M - j_{T,M,\min} H_{T,M}), \quad (4)$$

где $C_{M'}(i, j)$ – цвет пиксела (i, j) текстуры M' .

III. Если в тайле T_G пиксели окрашены в различные цвета, это означает, что для построения тайла T_E необходимо брать данные как из текстуры M , так и из текстуры D , смешивая их по маске G . Тогда сформируем тайл T_D , как описано в I, и текстуру M' , рассмотренную в II. Результирующий цвет C_P^{res} каждого пиксела $P(i, j)$ из подготавливаемого тайла T_E вычислим как

$$C_P^{res} = C_P \cdot F_G + C_D \cdot (1 - F_G),$$

где F_G – числовой эквивалент цвета пиксела (i, j) из тайла маски T_G , C_D – цвет пиксела (i, j) из тайла T_D , C_P – цвет, рассчитанный из текстуры M' по формуле (4).

Итак, мы рассмотрели процесс подготовки одного тайла текстуры E . Далее опишем технологию потайлового синтеза E с использованием распределенной вычислительной среды, которая состоит из ряда компьютеров, объединенных в сеть.

4 СИСТЕМА ПАРАЛЛЕЛЬНОГО СИНТЕЗА СВЕРХБОЛЬШИХ ТЕКСТУР

Для повышения эффективности и скорости подготовки текстуры E вычисление всех ее тайлов T_E предлагается выполнять параллельно с помощью системы синтеза сверхбольших текстур, которая имеет структуру, представленную на рис. 1. В данную систему входит ряд вычислительных узлов (ВУ), управляющий компьютер (УК), файловый сервер (ФС) и хранилище данных, в котором размещены массивы тайлов исходных текстур M , D и G (см. раздел 3). Каждый ВУ представляет собой персональный компьютер, оборудованный современной видеокартой (с многоядерным графическим процессором) и соединенный с УК и ФС с помощью сети, поддерживающей двусторонний обмен данными. Первоначально количество ВУ в системе равно $N_{ВУ}$, однако в процессе работы оно может уменьшаться в случае отказа/отключения отдельных узлов. Хранилище данных имеет постоянное соединение с файловым сервером по высокоскоростной шине. Рассмотрим принцип работы данной системы.

Запуск и последующий контроль синтеза сверхбольшей текстуры осуществляется с помощью управляющего компьютера. В задачу УК входит отправка вычислительным узлам заданий (очередь $Q_{заданий}$), содержащих номера тайлов текстуры E , которые необходимо подготовить, а также получение от каждого задействованного узла отчета (очередь $Q_{отчет}$) с кодом результата обработки задания и временем $t_{обп.}$, затраченным на его обработку. На основе поступающих от узлов отчетов УК формирует список тайлов текстуры E , подготовку которых не удалось выполнить в результате сбоя в системе.

Этот список обрабатывается отдельно по завершении работы системы. В нем фиксируются сбои трех типов. Сбой первого типа информирует о том, что задание на ВУ не может быть выполнено ввиду отсутствия в хранилище данных необходимых тайлов исходных текстур D или G . Сбой второго типа означает, что выполнение задания на ВУ было прервано из-за отказа/отключения ВУ или потери связи ВУ с УК, и задание может быть автоматически перезапущено на другом ВУ. К третьему типу относятся все оставшиеся сбои, которые разрешаются путем индивидуального анализа в ручном режиме.

Для определения порядка использования вычислительных узлов на УК размещается массив $A_{ВУ}$ размера $N_{ВУ}$, каждый элемент которого соответствует своему ВУ и хранит время $t_{обр.}$, затраченное этим узлом на обработку последнего исполненного задания (вначале $t_{обр.} = 0$), а также флаг, фиксирующий занятость ВУ (узел считается занятым с момента получения им задания до момента получения от него отчета). Обновление элементов массива $A_{ВУ}$ выполняется на основе данных из отчетов, поступающих от соответствующих узлов. В качестве следующего потенциального кандидата на получение задания УК выбирает первый свободный ВУ, имеющий наименьшее время $t_{обр.}$. Если свободный ВУ по каким-то причинам стал недоступен и

не может принять задание от УК, то для данного узла в массиве $A_{ВУ}$ в переменную $t_{обр.}$ записывается время $t_{отказ}$ продолжительности условного отказа, установленное на этапе инициализации системы (флаг остается в состоянии «свободен»). В случае если отчет не поступает от занятого ВУ в пределах времени $t_{обр,макс}$, соответствующего максимальному времени, которое отводится узлу на выполнение задания ($t_{обр,макс} < t_{отказ}$), то в соответствующую ячейку массива $A_{ВУ}$ заносится значение $t_{обр.} = t_{обр,макс}$, а флаг остается в состоянии «занят». Если по истечении времени $t_{отказ}$ УК не получает отчет от данного узла, то в ячейке массива $A_{ВУ}$ записывается время $t_{обр.} = t_{отказ}$, и флаг переводится в состояние свободен. Все узлы с $t_{отказ}$ по-прежнему участвуют в процессе распределения заданий, однако периодичность их опроса увеличивается до $t_{отказ}$. Таким образом, общая скорость синтеза текстуры E повышается за счет того, что вычислительные узлы, которым удается более быстро

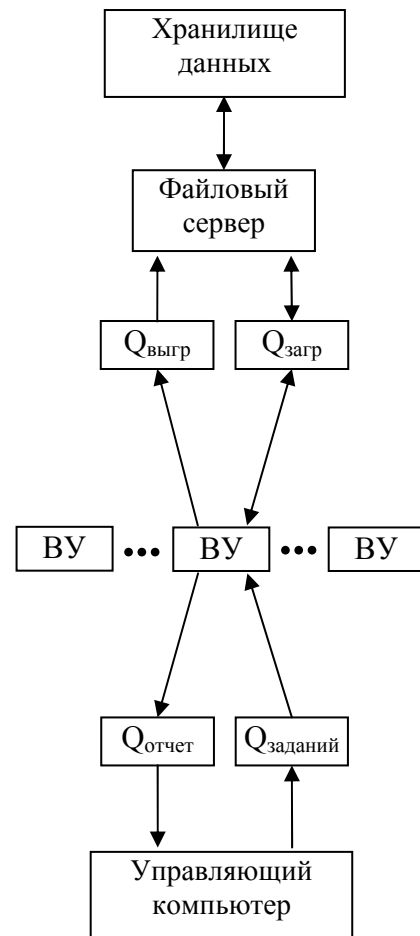


Рис. 1. Схема системы синтеза текстуры E

выполнить обработку заданий, используются чаще, чем более медленные узлы. При этом система сохраняет возможность автоматического включения в работу временно отказавших узлов.

Ключевым элементом системы, который выполняет непосредственно синтез тайлов текстуры E , является вычислительный узел. По номерам тайла текстуры E из задания, полученного от УК, вычислительный узел определяет необходимые номера тайлов маски G с учетом «обрамления» (см. выражение (2) из раздела 3) и отправляет запрос на их загрузку файловому серверу (очередь $Q_{\text{загр}}$). Получив необходимые тайлы текстуры G , узел формирует из них тайл T_G маски путем соответствующего масштабирования и обрезания и на основе T_G принимает решение о необходимости загрузки тайлов из текстур M и D (согласно разделу 3). В случае успешной загрузки всех необходимых тайлов ВУ выполняет синтез тайла T_E , выгружает его с помощью файлового сервера в хранилище данных (очередь $Q_{\text{выгр}}$), и отправляет отчет об успешном выполнении задания на УК (очередь $Q_{\text{отчет}}$). Отметим, что в случае отсутствия в хранилище данных необходимого тайла текстуры M , файловый сервер возвращает ВУ сгенерированный тайл черного цвета. В противном случае ВУ лишь отправляет на УК отчет с кодом ошибки (отказ файлового сервера, отсутствие в хранилище данных тайлов текстур G или D). Для сокращения времени подготовки тайлов обработка тайлов маски G и синтез тайла T_E текстуры E выполняются на GPU видеокарты, входящей в состав ВУ, с помощью архитектуры параллельных вычислений CUDA. Более подробно данная технология описана в работе².

Описанная система может быть развернута в рамках объединенных в локальную сеть персональных компьютеров, решающих различные задачи, и функционировать в фоновом режиме. В этом случае при увеличении вычислительной загруженности отдельных узлов другими, более приоритетными задачами, подготовка тайлов текстуры E автоматически перераспределяется между более свободными ВУ (вплоть до работы на одном ВУ), однако в этом случае общее время подготовки текстуры возрастает.

5 ЗАКЛЮЧЕНИЕ

Предложенная в данной статье технология позволяет синтезировать сверхбольшие текстуры поверхности планет по данным из нескольких исходных источников-текстур, имеющих различные картографические проекции, размеры и пространственные разрешения. Формирование результирующей текстуры выполняется по частям (тайлам), обрабатываемым независимо друг от друга, при этом обеспечивается бесшовная стыковка полученных фрагментов. Описанная в работе система синтеза сверхбольших текстур позволяет повысить общую скорость подготовки результирующей текстуры путем распараллеливания подготовки ее тайлов на отдельных вычислительных узлах (компьютерах). Описанная система обладает хорошей масштабируемостью и автоматически подстраивается под заданное количество узлов с учетом их вычислительной нагрузки. Работа выполнена при финансовой поддержке РФФИ в рамках гранта № 14-07-31332.

REFERENCES

- [1] Timokhin P.Yu., Mikhaylyuk M.V., “Sverkhbol'shie tekstury dlya vysoko realistichnoy vizualizatsii virtual'nykh landshaftov”, *Informatsionnye tekhnologii i vychislitel'nye sistemy*, **3**, 17-25 (2013).
- [2] Maltsev A.V., “Tekhnologiya formirovaniya detal'noy tekstury zemnoy poverkhnosti dlya kosmicheskikh trenazhernykh kompleksov”, *Informatsionnye tekhnologii*, **12**, 48-52 (2014).
- [3] Osborne P., “The Mercator projections”, *Virgin Media Inc., Edinburgh*, (2013), available at: <http://www.mercator99.webspace.virginmedia.com/mercator.pdf> (review date: 17.03.2015).

Поступила в редакцию 10.01.2015.